Brief Description of the Drawings

Fig. 1 is a schematic block diagram of a remote control unit and several consumer electronic devices arranged as an entertainment center.

Fig. 2 shows an illustrative grouping of enumerated function buttons into function classes.

Fig. 3 is a flowchart showing the operations in a process in accordance with the present invention.

Fig. 4 is a flowchart showing the operations in a process in accordance with the present invention.

Fig. 5 is a block diagram showing a system architecture including a schematic representation of a set of input buttons on a universal remote control unit, and several tables contained in the memory of such a universal remote control unit that may be used for classifying a user input within a particular function class and generating a classification code, determining which device is associated with the classification code and accessing the control code or codes needed to send the command indicated by a button press to the device which has a programmed association with the function class to which the button belongs.

Figs. 6A, 6B, and 6C are flowcharts of a process for setting up and utilizing, the programmed association feature in a specific embodiment of the present invention.

Fig. 7 is a flowchart of a process for disabling the programmed association feature in a specific exemplary embodiment of the present invention.

Fig. 8 is a flowchart of a process for enabling the programmed association feature in a specific exemplary embodiment of the present invention.

provides one or more signals which are used to access the appropriate Volume Change control codes for the TV. The ability to change the contents of target table **510** so that other devices may be associated with any particular function class is one aspect of the programmed association feature of the present invention.

In some embodiments of the present invention, the programmed association feature is implemented in software. Various software implementations may also be referred to as firmware. References to software or firmware each refer to having a stored control program executed by associated hardware. When a button on a remote control unit in accordance with present invention is pressed, the signal or signals generated thereby are interpreted as those representing button presses that are intended to transmit a message to the device(s) under control and those representing button presses that are not intended to transmit a message to the device(s) under control. Button presses that are intended to transmit a message are further classified, typically via a table lookup operation, as one of five classes, i.e., buttons that control volume; buttons that control channel selection; buttons that control power; buttons that control transport; and buttons that control other miscellaneous functions. If the programmed association feature is active (and not bypassed) this classification is used as an index into a table with one entry per possible classification. The table contains the starting context or target that the remote control unit should use when buttons of that class are pressed. Contexts, as used herein, are defined to be the type of device that is to receive commands from the remote control unit. Such devices include but are not limited to TVs and cable-boxes.

Example of Programmed Association Operation

Figs. 6A, 6B, and 6C are flowcharts of a process for setting up, and utilizing the programmed association feature in a specific exemplary

embodiment of the present invention. More particularly, and referring to Fig. 6A, this process is described beginning with a processor within the universal remote control unit being put into a low-power sleep mode **601**. A button press is then detected **602**. A keypad scanning function **603** generates the signals needed to determine what button was pressed. A decision is made **604** as to whether the program button was pressed or a button requiring the transmission of a command was pressed. If a button requiring transmission of a command was pressed, then a decision is made **605** as to whether on the numeric buttons (i.e., 0-9) was pressed. If a numeric button was pressed then a decision is made **606** as to whether the Program Sequence is equal to one of the values 2, 3, or 4. If the Program Sequence is equal to one of the values 2, 3, or 4, then software is executed **607** that increments the Program Sequence, and updates the DeviceID variable such that the new DeviceID value becomes equal to the previous DeviceID value multiplied by ten, plus the decimal value of the numeric key that was pressed. Additional software is executed **608** that displays on an LCD the digit of the numeric key that was pressed, moves the cursor displayed on the LCD, and flashes the LED one time. The processor then returns to the sleep mode **609**.

If at decision **604** it is determined that the program button was pressed, then one or more control flow instructions are executed **610** to go to software for the specific program function, as is illustrated more fully in Fig. 6C. Referring now to Fig. 6C, the description begins with the program button having been pressed **636**. A decision is then made **637** as to whether the Program Sequence is equal to the value 5. If it is, then the DeviceID variable is saved **644**; the text "Device Saved" is displayed on the LCD and the LED is flashed three times **645**; and the processor returns to sleep mode **646**. If at decision **637** the Program Sequence is not equal to the value 5, then a decision is made **638**, as to whether the Program Sequence is equal to the value 1. If it is, then, at **641**, the DeviceID variable is set to the value 0, and

the Program Sequence is set to the value 2. The LCD displays the text "Dev ID _" and the LED is flashed twice **642**. The processor then returns to sleep mode **643**. If at decision **638**, the Program Sequence was not equal to the value 1, then a decision is made **639** as to whether the Program Sequence is equal to the value 3. If it is, then various system adjustments and display functions may be executed **640**. If, at **639**, the Program Sequence is not equal to the value 3, then the Program Sequence is set equal to zero **647**, and the processor returns to sleep mode **648**.

Referring now to Fig. 6A, if at decision **605**, it is determined that a numeric button has not been pressed, then a decision is made **611**, as to whether the Program Sequence is equal to the value 2. If it is, then software is executed **618** that sets the value of the variable Class to the result of the Classify function that is executed with an argument indicative of which button was pressed. Fig. 2 shows a high level representation of exemplary table that can be used by the Classify function. That is, the information representative of the button that was pressed is compared to the list of buttons in the partitioned list shown in Fig. 2. When a match is found between the specific button that was pressed, and the button codes in the partitioned table, then a value is returned which represents the function class to which the button belongs. As shown at **619** in Fig. 6A, the Program Sequence variable is set to the value zero, the state of the universal remote control unit is set such that the programmed association feature is enabled, and a programmed association array is set so as to associate the selected function class with the current context, i.e., the presently selected A/V device source (a.k.a., the target of the control commands for that function class). Setting the state of the universal remote control unit is typically achieved by setting the binary state of one or more storage bits to a predetermined value. Additional software is executed **620**, such that the LED is flashed three times, and text is displayed on the LCD indicating the Priority (i.e., the programmed